

Kaltura Video Extension for Oracle Eloqua 10 Internal Information Guide

Version: 1.0

Kaltura Business Headquarters

250 Park Avenue South, 10th Floor, New York, NY 10003

Tel.: +1 800 871 5224

Copyright © 2014 Kaltura Inc. All Rights Reserved. Designated trademarks and brands are the property of their respective owners.
Use of this document constitutes acceptance of the Kaltura Terms of Use and Privacy Policy.

Contents

Setting up KMS for Eloqua.....	4
Creating an Eloqua Catalog App	4
Setting up Players	5

Setting up KMS and creating an Eloqua Catalog App for Deployment on the Customer's Eloqua's Instance

This information is intended for project managers and the professional services team that are deploying and providing services for the Kaltura Video Extension for Oracle Eloqua 10..

Setting up KMS for Eloqua

Who should perform this?

Someone that has access to KMS admin console.

To set up KMS for Eloqua

1. In KMS admin console, either create a KMS instance with an Eloqua profile for the customer's partner ID or add the Eloqua module to the customer's KMS.
2. In the customer's KMS admin:
 - a. Enable the Eloqua module.
 - b. Setup the players.
 - c. Make sure that the instance is on HTTPS.
 - d. Make sure anonymous is allowed.

Creating an Eloqua Catalog App

Who should perform this?

Someone who has access to our Eloqua developer instance and has done it before.

Recommended: Yulia, Gonen or Zohar

To create an Eloqua Catalog App

1. Go to our Eloqua developer instance.
2. Click 'Settings' and choose 'AppCloud Developer'.
3. Open in another tab the App called '[Kaltura Video Embed](#)' for reference.
4. Click Create new App.
5. Enter the following details:
 - a. Icon Url: https://site.kaltura.com/rs/kaltura/images/Kaltura_Logo_Medium.jpg
 - b. Name: Kaltura Video Embed

- c. Description: Embed a video
- d. Short description: Embed a video
- e. Enable URL: Concatenate the customer's KMS URL (in https) and "/eloqua/enable".
For example: <https://1777401.mediaspace.kaltura.com/eloqua/enable>
- f. Status URL: Concatenate the customer's KMS URL (inHTTPS) and "/eloqua/status".
For example: <https://1777401.mediaspace.kaltura.com/eloqua/status>
- g. OAuth Callback URL: Concatenate the customer's KMS URL (in HTTPS) and "/eloqua/configurationurl".
For example: <https://1777401.mediaspace.kaltura.com/eloqua/configurationurl>
6. Click Save.
7. Click Add Service.
8. Choose Content.
9. Enter the following details:
 - a. Icon URL: https://site.kaltura.com/rs/kaltura/images/Kaltura_Logo_Medium.jpg
 - b. Name: Kaltura Video Embed
 - c. Description: Embed a video
 - d. Create URL: Concatenate the customer's KMS URL (in HTTPS) and "/eloqua/createinstance". For example:
<https://1777401.mediaspace.kaltura.com/eloqua/createinstance>
 - e. Configure URL: Concatenate the customer's KMS URL (in HTTPS) and "/eloqua/configurationurl/instance/{InstanceId}". For example:
<https://1777401.mediaspace.kaltura.com/eloqua/configurationurl/instance/{InstanceId}>
 - f. Delete URL: Concatenate the customer's KMS URL (in HTTPS) and "/eloqua/removeinstance/instance/{instanceId}". For example:
<https://1777401.mediaspace.kaltura.com/eloqua/removeinstance/instance/{instanceId}>
 - g. Default content: content
 - h. Max Records per Notification: 100
 - i. Check 'Landing Pages' under 'This Service is available for use with:'. Do not check 'Emails'
 - j. Landing Page Notification (HTML) URL: Concatenate the customer's KMS URL (inHTTPS) and "/eloqua/renderinstance/instance/{InstanceId}". For example:
<https://1777401.mediaspace.kaltura.com/eloqua/renderinstance/instance/{InstanceId}>
10. Click Save.
11. Click Return to App.
12. On the bottom of the page copy the URL under the Catalog. This is what should be sent to the customer.
It is recommended to test that it works.

Setting up Players

To add the Eloqua reporting plugin to a player

1. In the UIConf config parameter, under the plugin section add :

```
"eloquaReport": {  
  "analyticsFormUrl": null,  
  "analyticsFormName": null,  
  "analyticsSiteId": null,  
}
```

Setting up KMS and creating an Eloqua Catalog App for Deployment on the Customer's Eloqua's Instance

```
"milestones": "1,5,10,25,50,75,99,100",  
"milestoneInterval": 30,  
"lookupKey": null,  
"emailFieldName": "ElqEmailAddress"  
}
```

Where:

- k. The first 3 parameters you get from the customer from the defined form for video engagement statistics
- l. "Milestones" are predefined points in percentage watched where the player sends a report. This is useful for short videos.
- m. "MilestoneInterval" is the amount in seconds between player reports (in addition to the percentage milestones). This is useful for longer videos.
- n. "lookupkey" is the lookup key for the email field in the customer's Eloqua instance
- o. "emailFieldName" is the name of the email address field (default name is "ElqEmailAddress")

To add a data collection form to a player

1. In the UIConf config parameter, under the plugin section add :

```
"actionForm": {  
  "displayOn": "start", // start, <time>, <percent>%, end  
  "displayOnce": true,  
  "submitRequired": false,  
  "description": "For more information, please enter your details and we will get  
back to you",  
  "fields": [  
    {  
      "name": "name",  
      "placeholder": "Name",  
      "type": "text",  
      "required": true  
    },  
    {  
      "name": "email",  
      "placeholder": "Email",  
      "type": "email",  
      "required": true  
    },  
    {  
      "name": "phone",  
      "placeholder": "Phone Number",  
      "type": "text"  
    }  
  ],  
  "templatePath": "../CallToAction/templates/collect-form.tmpl.html"  
}
```

Where:

- a. "displayOn" is when the form is presented in the video. Acceptable values are:
 - "start" – before playback
 - "end" – after playback
 - A number – the amount of seconds from the start of the playback
 - A percentage – the percentage of the video played
- b. "displayOnce" has not been developed and should be ignored.
- c. "submitRequired" is whether there will be a 'skip' option. Valid values are true or false

- d. "description" is the text appearing before the fields
- e. For each field:
 - o "name" should be exactly as the field name on the Eloqua form
 - o "placeholder" is the text that appears in the field before filling it out
 - o "type" is the field type. It is recommended to keep to: text, email, number. Valid values are theoretically every valid input type in HTML, as defined here: http://www.w3schools.com/tags/att_input_type.asp. Practically, some won't work properly because they need a value set (e.g. hidden, radio) and some types are not supported across all browsers. We have tested the following types: text, email, number. Every other type should be tested to see if it works properly before communicating its availability to the customer.
 - o "required" – whether it is a required field

2. In the UIConf config parameter, under the plugin section add :

```
"eloquaReport": {  
  "collectionFormUrl": null,  
  "collectionFormName": null,  
  "collectionSiteId": null,  
  "lookupKey": null,  
  "emailFieldName": "ElqEmailAddress"
```

Where:

- a. The first 3 parameters you get from the customer from the defined form for this data collection form.
- b. "lookupkey" is the lookup key for the email field in the customer's Eloqua instance.
- c. "emailFieldName" is the name of the email address field (default name is "ElqEmailAddress").

To add Call-To-Actions to a player

1. In the UIConf config parameter, under the plugin section add :

```
"actionButtons": {  
  "displayOn": "end", // Can be either "end" or "related"  
  "customDataKey": "ActionButtons",  
  "openInNewWindow": true,  
  "actions": [  
    {"id": "product", "label": "Buy this Product", "url":  
      "http://www.myproduct.com"},  
    {"id": "product2", "label": "Read more about this Product", "url":  
      "http://www.myproduct.com"}  
  ],  
  "templatePath": "../CallToAction/templates/action-buttons.tmpl.html"  
}
```

Where:

- a. "displayOn" should be "end" if there is no related
- b. "customDataKey" is the name of the custom data field in which a CTA override can be added.



NOTE: If there is more than one custom data profile/scheme defined for the partner ID, make sure that UIVar includes the parameter "metadataProfileId" with the value of the Id of the relevant custom data profile profile/scheme.

- c. "openInNewWindow" is whether clicking the button opens in the same tab/window as the player or a new one

- d. For each CTA:
 - "id" is the button ID used for reporting
 - "label" is what appears on the label
 - "url" is the URL that clicking the button leads to



NOTE: If there is more than one custom data profile/scheme defined for the partner ID, make sure that UIVar includes the parameter "metadataProfileId" with the value of the Id of the relevant custom data profile profile/scheme.

To add Call-To-Actions with related to a player

1. Make sure the player has the related plugin set on.
2. In the UIConf config parameter, under the plugin section add:

```
"actionButtons": {
  "displayOn": "related", // Can be either "end" or "related"
  "customDataKey": "ActionButtons",
  "openInNewWindow": true,
  "actions": [
    {"id": "product", "label": "Buy this Product", "url":
"http://www.myproduct.com"},
    {"id": "product2", "label": "Read more about this Product", "url":
"http://www.myproduct.com"}
  ],
  "templatePath": "../CallToAction/templates/action-buttons.tmpl.html"
},
"related": {
  "itemsLimit": 5
}
},
```

Where:

- a. "displayOn" should be 'related'
- b. Other parameters explained above
- c. "Related: { "itemsLimit" ... }" is how many maximum items to display in the related.

To change the look and feel of all the above visible items in a player

1. Create a CSS file with all the desired changes and host it in an HTTP accessible site.
Note that:

- a. In general, you can find the CSS types by using inspect element
- b. CSS attributes of CTA buttons are under:

```
.cta-button {
```

- c. Fields in a data collection form are under:

```
.cta-form-item input {
```

- d. The text above the fields in a data collection form is under:

```
.actionForm h3 {
```


Setting up KMS and creating an Eloqua Catalog App for Deployment on the Customer's Eloqua's Instance

e. The submit button for a data collection form is under:

```
input[type="submit"] {
```

f. For each CSS attribute add "important" at the end of the line, to make sure it overrides

See examples in:

- o <http://demo.kaltura.com/users/oferr/connect2014/BlazitorCTA1Skin.css>
- o <http://demo.kaltura.com/users/oferr/connect2014/BlatizorCTASkin.css>

2. In the UIConf config parameter, under the plugin section add under layout a parameter called "cssFiles" with a URL for the CSS file, like so:

```
"layout": {  
  "skin": "kdark",  
  "cssFiles": [  
    "http://demo.kaltura.com/users/oferr/connect2014/BlazitorCTA1Skin.css"  
  ]  
}
```



NOTE: The intent is to integrate the kaltura Video Extension for Oracle Eloqua plugin with the Universal Studio and v2 Player.